What is claimed is:

1. A graphics rendering software program for providing instructions to one or more processors to render graphics on a display of an embedded computing device configured for establishing a network connection with at least one other computing device, comprising:

(a) an application layer;

(b) a graphics toolkit; and

(c) a graphics driver for rendering a plurality of drawing surfaces, including a first drawing surface and a second drawing surface, on a display of the embedded computing device, and

(d) wherein the graphics driver is configured to render the first drawing surface at least partially overlapping the second drawing surface on the display, and

(e) wherein, when the first drawing surface is rendered as partially overlapping the second drawing surface, a visible portion of the second drawing surface is computed as a set of rectangular clip segments.

2. The graphics rendering software program of Claim 1, wherein the graphics driver is further configured to increment a visibility tag corresponding to second drawing surface when the visible portion of the second drawing surface is changed.

3. The graphics rendering software program of Claim 2, wherein the graphics driver is further configured to compute a new set of rectangular clip segments when the visible portion of the second drawing surface is changed.

4. The graphics rendering software program of Claim 1, wherein the first drawing surface and the second drawing surface, including both the visible portion and an obscured portion, comprise rectangular borders.

5. The graphics rendering software program of Claim 1, wherein each rectangular clip segment of the set of rectangular clip segments is iteratively output to the display for displaying the visible portion of the second drawing surface.

6. A graphics rendering software program for providing instructions to one or more processors to render graphics on a display of an embedded computing device configured for establishing a network connection with at least one other computing device, comprising:

(a) an application layer;

(b) a graphics toolkit; and

(c) a graphics driver for rendering a plurality of drawing surfaces, including a first drawing surface and a second drawing surface, on a display of the embedded computing device, and

(d) wherein the graphics driver is configured to render the first drawing surface at least partially overlapping the second drawing surface on the display, and

(e) wherein, when the first drawing surface is rendered as partially overlapping the second drawing surface, a visible portion of the second drawing surface is computed as a set of rectangular clip segments, and

(f) wherein the set of rectangular clip segments is stored as a graphics context object corresponding to unobscured segments of the second drawing surface.

7. The graphics rendering software program of Claim 6, wherein the graphics driver is further configured to increment a visibility tag corresponding to second drawing surface when the visible portion of the second drawing surface is changed.

8. The graphics rendering software program of Claim 7, wherein the graphics driver is further configured to compute a new set of rectangular clip segments when the visible portion of the second drawing surface is changed.

9. The graphics rendering software program of Claim 6, wherein the first drawing surface and the second drawing surface, including both the visible portion and an obscured portion, comprise rectangular borders.

10. A graphics rendering software program for providing instructions to one or more processors to render graphics on a display of an embedded computing device configured for establishing a network connection with at least one other computing device, comprising:

(a) an application layer;

(b) a graphics toolkit; and

(c) a graphics driver for rendering a plurality of drawing surfaces, including a first drawing surface and a second drawing surface, on a display of the embedded computing device, and

(d) wherein the graphics driver is configured to render the first drawing surface at least partially overlapping the second drawing surface on the display, and

(e) wherein, when the first drawing surface is rendered as partially overlapping the second drawing surface, a visible portion of the second drawing surface is computed as a set of rectangular clip segments, and

(f) wherein the set of rectangular clip segments is stored as a graphics context object corresponding to unobscured segments of the second drawing surface, and

(g) wherein each rectangular clip segment of the set of rectangular clip segments is iteratively output to the display for displaying the visible portion of the second drawing surface.

11. The graphics rendering software program of Claim 10, wherein the graphics driver is further configured to increment a visibility tag corresponding to second drawing surface when the visible portion of the second drawing surface is changed.

12. The graphics rendering software program of Claim 11, wherein the graphics driver is further configured to compute a new set of rectangular clip segments when the visible portion of the second drawing surface is changed.

13. The graphics rendering software program of Claim 10, wherein the first drawing surface and the second drawing surface, including both the visible portion and an obscured portion, comprise rectangular borders.

14. A graphics rendering software program for providing instructions to one or more processors to render graphics on a display of an embedded computing device configured for establishing a network connection with at least one other computing device, comprising:

(a) an application layer;

(b) a graphics toolkit; and

(c) a graphics driver for rendering a plurality of drawing surfaces, including a first drawing surface and a second drawing surface, on a display of the embedded computing device, and

(d) wherein the graphics driver is configured to render the first drawing surface at least partially overlapping the second drawing surface on the display, and

(e) wherein, when the first drawing surface is rendered as partially overlapping the second drawing surface, a visible portion of the second drawing surface is computed as a set of clip segments.

15. The graphics rendering software program of Claim 14, wherein the graphics driver includes:

(1) a shape function layer including a target architecture specific instruction set for setting and retrieving pixel values, respectively, into and from a one-dimensional framebuffer memory; and

(2) a framebuffer access macro layer including a set of macros for inlining into the shape function layer.

16. A graphics rendering software program for providing instructions to one or more processors to render graphics on a display of an embedded computing device configured for establishing a network connection with at least one other computing device, comprising:

(a) an application layer;

(b) a graphics toolkit; and

(c) a graphics driver for rendering a plurality of drawing surfaces, including a first drawing surface and a second drawing surface, on a display of the embedded computing device, and

(d) wherein the graphics driver is configured to render the first drawing surface at least partially overlapping the second drawing surface on the display, and

(e) wherein, when the first drawing surface is rendered as partially overlapping the second drawing surface, a visible portion of the second drawing surface is computed as a set of clip segments, and

(f) wherein the set of clip segments is stored as a graphics context object corresponding to unobscured segments of the second drawing surface.

17. The graphics rendering software program of Claim 16, wherein the graphics driver includes:

(1) a shape function layer including a target architecture specific instruction set for setting and retrieving pixel values, respectively, into and from a one-dimensional framebuffer memory; and

(2) a framebuffer access macro layer including a set of macros for inlining into the shape function layer.

18. A graphics rendering software program for providing instructions to one or more processors to render graphics on a display of an embedded computing device configured for establishing a network connection with at least one other computing device, comprising:

(a) an application layer;

(b) a graphics toolkit; and

(c) a graphics driver for rendering a plurality of drawing surfaces, including a first drawing surface and a second drawing surface, on a display of the embedded computing device, and

(d) wherein the graphics driver is configured to render the first drawing surface at least partially overlapping the second drawing surface on the display, and

(e) wherein, when the first drawing surface is rendered as partially overlapping the second drawing surface, a visible portion of the second drawing surface is computed as a set of clip segments, and

(f) wherein the set of clip segments is stored as a graphics context object corresponding to unobscured segments of the second drawing surface, and

(g) wherein each clip segment of the set of clip segments is iteratively output to the display for displaying the visible portion of the second drawing surface.

19. The graphics rendering software program of Claim 18, wherein the graphics driver inlcudes:

(1) a shape function layer including a target architecture specific instruction set for setting and retrieving pixel values, respectively, into and from a one-dimensional framebuffer memory; and

(2) a framebuffer access macro layer including a set of macros for inlining into the shape function layer.

20. A method of rendering graphics including overlapping drawing surfaces on a display of an embedded computing device configured for establishing a network connection with at least one other computing device, comprising the steps of:

(a) computing a set of clip segments corresponding to a visible portion of a partially obscured drawing surface; and

(b) rendering the partially obscured drawing surface along with an overlapping drawing surface on the display.

21. The method of Claim 20, wherein the clip segments correspond to rectangular portions of the visible portion of the partially obscured drawing surface.

22. The method of Claim 21, wherein the partially obscured drawing surface and the overlapping drawing surface comprise rectangular borders.

23. The method of Claim 21, further comprising the step of storing the set of clip segments as a graphics context object corresponding to unobscured segments of the partially obscured drawing surface.

24. The method of Claim 23, wherein the clip segments correspond to rectangular portions of the visible portion of the partially obscured drawing surface.

25. The method of Claim 24, wherein the partially obscured drawing surface and the overlapping drawing surface comprise rectangular borders.

26. The method of Claim 23, further comprising the step of iteratively outputting each clip segment of the set of clip segments to the display for displaying the visible portion of the partially obscured drawing surface.

27. The method of Claim 26, wherein the clip segments correspond to rectangular portions of the visible portion of the partially obscured drawing surface.

28. The method of Claim 27, wherein the partially obscured drawing surface and the overlapping drawing surface comprise rectangular borders.